TCP-LP: Fair and Friendly Congestion Control Approach

Deepak Mehta[1]
[1](Research Scholar, Dept. of CSA, JJT University, Rajasthan, India)

ABSTRACT : *Internet users always seek service prioritization. This service can be defined as"Give importanceto important network traffic over unimportant network traffic".Conventional methods can be categorization of traffic by considering the existing traffic as"best-effort" class can be named as low-priority (LP) class, and keen to develop mechanisms which will give "better-than-best-effort" service. It is worth mentioning thatthis paper is going to mention the ides of developing a Low Priority distributed algorithm whose objective is to utilize only the rest bandwidth and give priority to other delay sensitive traffic generated by interactive applications or media streaming application and interested in devising a efficient approach or algorithm called novel distributed algorithm to implement a LP service which work as against existing best effort traffic service from the communication endpoints.*

Keywords-*AQM, drop-tail, fairness, reno, SFQ, vegas.*

I. Introduction

In most of network large portion of bandwidth remain unused when accessing the network. An Internet user always wants to use this unused extra bandwidth in other network enabled applications without affecting the traffic that is already in use. Internet applications can be categorized into two broad categories named Delay sensitive application and Non Delay Sensitive applications.Background applications which are not sensitive to delay can be softwareupdates or can be file-sharing applications such as peer to peer service. But TCP doesn't have abilities which can differentiate Internet traffic into among classes. TCP always distribute the bandwidth evenly the connections .Therefore P2P application like Bit Torrent made multiple TCP Connections to download or upload the data.

Purposed Low Priority algorithm (TCP-LP) is delay based congestion control algorithm which is experimented on NS2.35 simulator which can be characterized asreactive to earlycongestion in the network, thus enable background traffic to use the services of network while avoiding interference or give priority over them.TCP-LP always takes three considerations in case congestion avoidance. First objective is to quickly reduce the sending rate by decreasing the size of congestion window in the presence of congestion from TCP flows. Second objective is to quickly utilize the rest bandwidth in absence of sufficient traffic and last important objective to achieve intera-protocol fairness among multiple flows of TCP-LP. To completely fulfill these objectives this paper purposed to modifies the conventional used AIMD congestion avoidance policy with an inference stage and also purposed modification in back off policy.

II. Related Work

Ana Hernandez et. al [1] focus on the use of one way delay (OWD) as compare to use of Round Trip Time (RTT). Author emphasized that it provide much more information. TCP-LP use OWD as a tool to calculate window size dynamically. Brian Ottet. al [2] concluded lower priority traffic which operate in background exhibit high average packet delay which is unimportant for increasing goodput and also characterized by it more aggressive protocol.

Chi Wang in [3] claimed that priorities can't be divided into two fixed level of priorities. It depends on the user needs or depends on application demand. TCP-LP provides flexible level of priorities which can be manually set by user and even user can utilize the bandwidth on the basis of priorities.

C. Callegari [4] discussed and experimented 13 different type of congestion control protocols. TCP-LP , TCP-Reno and Vegas among of them. As post metric results exhibit TCP-LP always work in background allow TCP-Reno to generate more throughputs in wired and wireless environment.

A. Venkataramani [5][6] claimed about the relationship between TCP-LP and TCP-Nice. Both provide support for file replication in background and both protocol developed in parallel mode didn't know about each other when developed. L. Brakmo in [7] claimed that TCP-Nice is extended version of TCP –Vegas. New feature Congestion detector sensitivity is added in TCP Nice and uses RTT threshold based congestion detector for detecting the congestion. When more than 50% packet becomes victim of RTT delay then it is indication of congestion and TCP-Nice congestion sensitivity detector start working. If compare with TCP-LP , it uses one way delay for congestion detection and TCP Nice use RTT based threshold for congestion indication. As similar LEDBAT also use OWD scheme it means all low priority based congestion control protocol always give priority to other traffic when share the bottleneck bandwidth. TCP-LP when encounter congestion it decreases its congestion window size as described in [8].

III. PROBLEM FORMULATION

As explained earlier TCP-LP is sender side modified algorithm of TCP designed especially for low priority services. In this paper our goal is measure the performance of TCP-LP by simulating the wired environment in NS2.35 and also measure fairness and friendliness of existing congestion control (Reno, Vegas) protocol with TCP-LP on shared link.

IV. NS2 SIMULATOR

NS2 [9] [10] is used for research in network and it is basically discrete event simulator. With NS2 user can experiment the simulation of TCP, support simulation of routing protocols in wired and wireless network. As depicted in figure 1 an abstract view of NS2 which contain Object oriented TCP script interpreter that consist event simulation scheduler and also consist network component object library.

.

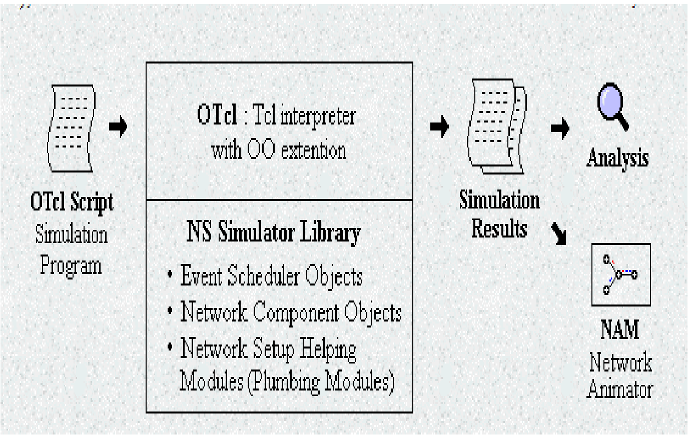

Fig.1: NS2 Abstract View

V. SIMULATED ENVIRONMENT WITH EXPERIMENTAL RESULT

In the below figure1 we present simulated environment with three source nodes 0, 1, 3 respectively with bottleneck bandwidth link 1.5 Mbps with RTT 100 MS and loss probability (Packet loss is measured as a percentage of packets lost with respect to packets sent) is fixed to 0.0001. Table 1 represents Nodes configuration with queue managements and their queue limits and also Table 2 represent Nodes with their associated TCP Agents in NS2.35 [9] in Cent OS 6.3.

TABLE I. Nodes Configuration

Node	Active Queue Management Technique	Queue Limit
Node 0, Node 1 to Router- 2	Drop Tail	50
Node 3 to Router-2	SFQ	50
Router-2 to Router-5	Drop Tail	12
Router-5 to Node 4	Drop Tail	50
Router-5 to Node 6	SFQ	50

Source Author

TABLE II. Nodes Traffic Generator Agents

Node	TCP Agents	Packet Size(K)	Connection Type (Traffic Generator)
Node 0	Reno	280	Short Lived Connection (Pareto Traffic)
Node 1	Vegas	210	Short Live connections (Exponentials Traffic)
Node 3	LP	1	Long lived connections (FTP Traffic)

Following figures 3, 4 and 5 represents congestion window behaviors of Node0 (Reno), Node 3 (LP) and Node 1 (Vegas).As below figures show congestion window maximum size is approx. 30.In case of Reno and LP, it starts sending more packets at slow start phase indicate every packet has successfully transmitted in one second (initially sender start the transmission by one segment and wait for acknowledgement and when acknowledgement received it increment the congestion window size by one and then send two packet and wait for acknowledgement and when acknowledgement received it increment again the congestion window by four therefore increment exponentially.) but in Vegas (Node 2) less packets transmitted as compare to other source nodes. After 1 second indication of packet loss its half the congestion window and go linearly and after that it came down to zero size in Reno this phase is called congestion avoidance phase and Fast Retransmit case it took the window by dynamically setting the threshold but TCP-LP behavior is good as good as compare to other nodes.

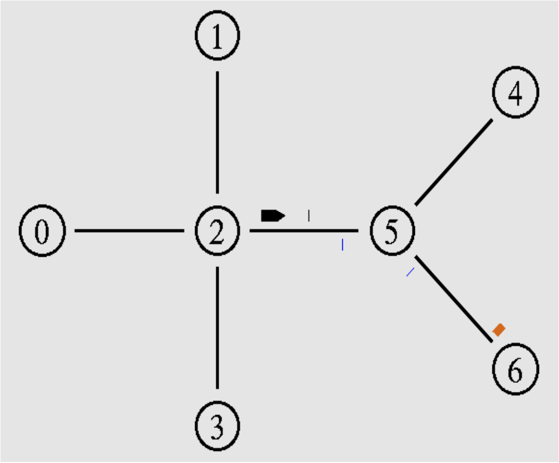

Fig. 2: Wired Environment (Source Author)

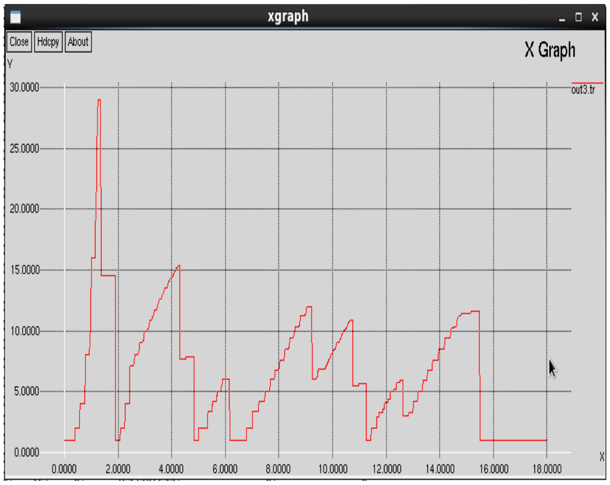

Fig. 3: Node0 CWND (Reno) Source Author

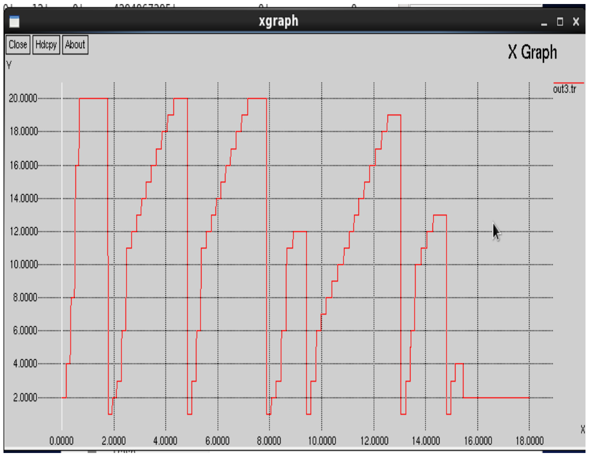

Fig. 4: Node3 CWND (LP) Source Author

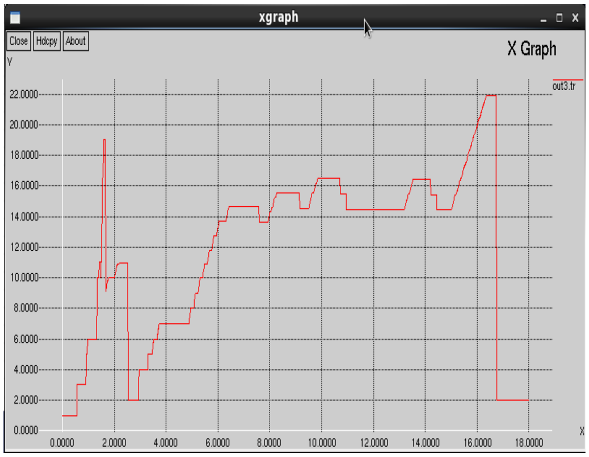

Fig. 5: Node1 CWND(Vegas)Source Author

TABLE III. Wired Performance Metrics

NODES WITH PROTOCOL	PDR	Throughput (Kbps)	Sys throughput	sent	Received
TCP_LP (NODE 3)	98.72%	377.194	1362.54	704	695
TCP-Vegas (NODE 1)	99.90%	717.102 (↑31.3% >Reno)		971	970
TCP-Reno (NODE 0)	97.58%	224.591		413	403

Source Author

Table 3 presents a performance metric after simulation results of above said scenario. As TCP-LP exhaust only rest bandwidth which is not used by other nodes and in our case it is very friendly and not harm other connection parallel to it and overall system throughput is best. As below fig. 6 represents comparison of instant throughput of Reno and LP and fig. 7 represent instant throughput comparisons of LP and Vegas during simulation.

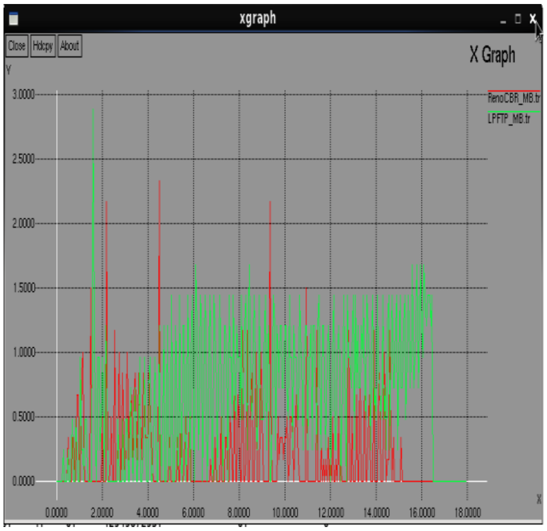

Fig. 6: Reno vs. LP throughputs (Source Author)

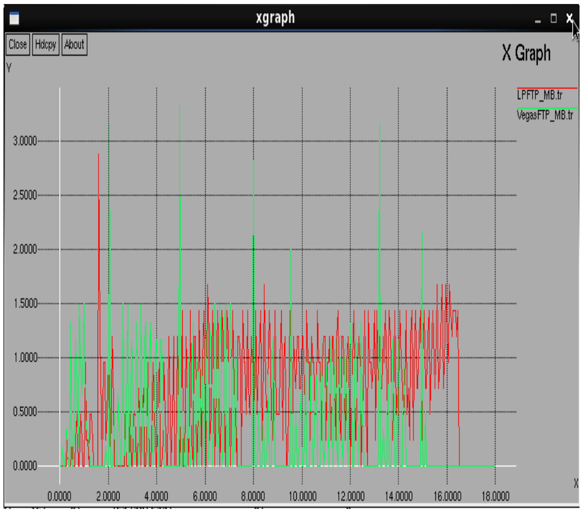

Fig. 7: LP vs. Vegas throughputs (Source Author)

As figure 5 reflects TCP-LP throughput with TCP Reno over 15 minutes simulation , it's always keep its throughput low whenever there is TCP-Reno active and gave priority to other therefore TCP-Reno got throughput higher as figure reflect. Whenever bandwidth is free TCP-LP speed its process of generating throughput.

Figure 6 depicts the same nature as above explained with TCP-Vegas;However, when at 7 minutes, 15 minutes to 16 minute, TCP-LP finds such opportunity to generate traffic and use all bottleneck bandwidth and successfully highest TCP throughput.

VI. FRIENDLINESS AND FAIRNESS

Fairness [11] refers to how TCP with different variation or flavors affect the output of one another. User always interested in developing protocol which should be fair and friendly to other existing protocols running simultaneous on the connection. A newly purposed connection should not make the cause of starvation for other protocols.

As above described simulation environment have N=1080 greedy connections with LP, Reno and Vegas connections. Jain Fairness index can be described as the ability of TCP of sharing the bottleneck capacity among the different sessions active on the bottleneck link. It is supposed all the connections are using the same TCP versions. The maximum level of fairness reached in the case when the N TCP session sharing the same bottleneck link gets 1/N of the link capacity each. In this simulated environment consider three competing connections.

A quantitative analysis of the fairness Jain index is used following formula is used

$$\mathcal{J}(x_1, x_2, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} x_i{}^2}$$

It gives the result between 0 and 1. One represents best fairness and zero represents very poor fairness.

TABLE IV. Performance Metric

No. of Connection with TCP Agent	Fairness Index	Mean Throughput(Kbps)
360 Reno	0.003	0.184
360 Reno **360 LP** **360 Vegas**	0.001	0.383
360 LP	0.002	0.646
360 Vegas	0.002	0.319

Source Author

Table 4 represents Jain Fairness index with Means throughputs achieved by the purposed protocols. Results show that fairness indexes which are obtained by protocols in the mixed simulation environments are better than ones obtained with only TCP -Reno protocols; also result concludes that TCP-LP is best among of them and very fair and friendly towards Reno and Vegas.

VII. CONCLUSION

Surely TCP is standard protocol for communication. Several TCP variants have been introduced to override the limitation of TCP. However Reno is standard protocol for congestion control but use of TCP-Vegas display exceptional output with 31.32% more in throughput as compare to TCP-Reno. Intuitively TCP-LP use rest bandwidth and allow other traffic to precede therefore its throughput averagely exhibit worst performance also represent fair and friendly approach towards Vegas and Reno.

REFERENCES

[1] A. Hernandez and E. Magana, "One-way delay measurement and characterization”, Proc. 3rd IEEE Conf. Networking and Services ICNS, 2007.

[2] B. Ott, T. Warnky and V. Liberatore, “Congestion Control for low Priority Traffic”, Proc. SPIE.5245, Internet Quality of Service, 154, 2003.

[3] C.Wang, H. Wang, Y. Lin, S.Chen and C. Wu, “Charge-based low priority congestion control”, Proc. 3rd IEEE Conf. 2008.

[4] C. Callegari, S. Giordano, M. Pagano and T. Pepe –“Behavior analysis of TCP Linux variants”, Computer Networks: The International Journal of Computer and Telecommunications Networking, 56(1), 2012, 462-476.

[5] Venkataraman, Arun, R. Kokku, and M. Dahlin, "TCP Nice: A mechanism for background transfers", ACM SIGOPS Operating Systems Review, 36, 2002, 329-343.

[6] Kuzmanovic, Aleksandar, and E. W. Knightly, "TCP-LP: A distributed algorithm for low priority data transfer", INFOCOM, Proc. Annual Joint IEEE Conf. Computer and Communication, 3, 2003.

[7] Brakmo, S. Lawrence and Larry L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet", IEEE Journal on Selected Areas in Communications, 13(8), 1995, 1465-1480.

[8] S. Floyd, "TCP and explicit congestion notification", ACM SIGCOMM Computer Communication Review 24(5), 1994, 8-23.

[9] http://www.isi.edu/nsnam/ns/

[10] http://nile.wpi.edu/NS/

[11] Norlund, Krister, T.Ottosson, and A. Brunstrom, "TCP fairness measures for scheduling algorithms in wireless networks", Proc. 2nd IEEE Conf. Quality of Service in Heterogeneous Wired/WirelessNetworks, 2005.